

mysql 触发器(Trigger)简明总结和使用实例

2014/11/18 29132 这篇文章主要介绍了 mysql 触发器(Trigger)简明总结和使用实例,需要的朋友可以参考下 一, 什么触发器 1, 个人理解 触发器, 从字面来理解, 一触即发的一个器, 简称触发器 (哈哈, 个人理解), 举个例子吧, 好比天黑了, 你开灯了, 你看到东西了。你放炮仗, 点燃了, 一会就炸了。 2, 官方定义 触发器 (trigger) 是个特殊的存储过程, 它的执行不是由程序调用, 也不是手工启动, 而是由事件来触发, 比如当对一个表进行操作 (insert, delete, update) 时就会激活它执行。触发器经常用于加强数据的完整性约束和业务规则等。 触发器可以从 DBA_TRIGGERS , USER_TRIGGERS 数据字典中查到。 触发器有一个非常好的特性就是:触发器可以禁止或回滚违反引用完整性的更改, 从而取消所尝试的数据修改。 什么意思, 举个例子解释一下, 街机游戏大家都玩过吧, 闯过一关, 闯下一关, 有一关没闯过就要从第一关开始。触发器跟这个类似。 官方解释如下 触发程序视为单一交易中的一部份, 因此可以由原触发程序还原交易, 如果在交易过程中侦测到严重的错误 (如使用者中断连线), 则会自动还原整个交易。 他的作用很明显了, 可以保重数据的完整性, 下面有一个实例来说明他的好处, 以及如果使编写代码不那么复杂 二, 触发器语法 CREATE TRIGGER trigger_name trigger_time trigger_event ON tbl_name FOR EACH ROW trigger_stmt 触发程序是与表有关的命名数据库对象, 当表上出现特定事件时, 将激活该对象。 触发程序与命名为 tbl_name 的表相关。tbl_name 必须引用永久性表。不能将触发程序与 TEMPORARY 表或视图关联起来。 trigger_time 是触发程序的动作时间。它可以是 BEFORE 或 AFTER, 以指明触发程序是在激活它的语句之前或之后触发。 trigger_event 指明了激活触发程序的语句的类型。trigger_event 可以是下述值之一: · INSERT: 将新行插入表时激活触发程序, 例如, 通过 INSERT、LOAD DATA 和 REPLACE 语句。 · UPDATE: 更改某一行时激活触发程序, 例如, 通过 UPDATE 语句。 · DELETE: 从表中删除某一行时激活触发程序, 例如, 通过 DELETE 和 REPLACE 语句。 请注意, trigger_event 与以表操作方式激活触发程序的 SQL 语句并不很类似, 这点很

重要。例如，关于 INSERT 的 BEFORE 触发程序不仅能被 INSERT 语句激活，也能被 LOAD DATA 语句激活。 可能会造成混淆的例子之一是 INSERT INTO .. ON DUPLICATE UPDATE ...语法：BEFORE INSERT 触发程序对于每一行将激活，后跟 AFTER INSERT 触发程序，或 BEFORE UPDATE 和 AFTER UPDATE 触发程序，具体情况取决于行上是否有重复键。 对于具有相同触发程序动作时间和事件的给定表，不能有两个触发程序。例如，对于某一表，不能有两个 BEFORE UPDATE 触发程序。但可以有 1 个 BEFORE UPDATE 触发程序和 1 个 BEFORE INSERT 触发程序，或 1 个 BEFORE UPDATE 触发程序和 1 个 AFTER UPDATE 触发程序。 trigger_stmt 是当触发程序激活时执行的语句。如果你打算执行多个语句，可使用 BEGIN ... END 复合语句结构。这样，就能使用存储子程序中允许的相同语句 三，创建解发器 1,用户表 user

```
CREATE TABLE `user` ( `id` int(11) NOT NULL auto_increment COMMENT '用户 ID', `name` varchar(50) NOT NULL default '' COMMENT '名称', `sex` int(1) NOT NULL default '0' COMMENT '0 为男, 1 为女', PRIMARY KEY (`id`) ) ENGINE=MyISAM DEFAULT CHARSET=utf8 ; INSERT INTO `user` (`id`, `name`, `sex`) VALUES (1, '张映', 0), (2, 'tank', 0); 2,评论表 comment CREATE TABLE `comment` ( `c_id` int(11) NOT NULL auto_increment COMMENT '评论 ID', `u_id` int(11) NOT NULL COMMENT '用户 ID', `name` varchar(50) NOT NULL default '' COMMENT '用户名称', `content` varchar(1000) NOT NULL default '' COMMENT '评论内容', PRIMARY KEY (`c_id`) ) ENGINE=MyISAM DEFAULT CHARSET=utf8 ; INSERT INTO `comment` (`c_id`, `u_id`, `name`, `content`) VALUES (1, 1, '张映', '触发器测试'), (2, 1, '张映', '解决字段冗余'), (3, 2, 'tank', '使代码更简单'); 在这里有一个冗余字段 name，我们在读取评论进可以用联合查寻来找到 user 表中的名字，为什么要有冗余字段呢，因简单的 sql 语句执行效率更高，但不是冗余字段越多越好，冗余字段多了，同样会增加数据库负担 。 我要做的事情是，当我更新 user 表
```

的 name 时，触发器同时更新 comment 表，就不要写 php 代码去更新了，当用户被删除时，comment 表中，有关该用户的数据将被删除 3，更新 name 触发器

```
delimiter || //mysql 默认结束符号是分号，当你在写触发器或者存储过程时有分号出现，会中止转而执行 drop trigger if exists updatename|| //删除同名的触发器， create trigger updatename after update on user for each row //建立触发器， begin //old,new 都是代表当前操作的记录行，你把它当成表名，也行; if new.name!=old.name then //当表中用户名称发生变化时,执行 update comment set comment.name=new.name where comment.u_id=old.id; end if; end|| delimiter ;4，触发器删除 comment 数据 delimiter || drop trigger if exists deletecomment|| create trigger deletecomment before delete on user for each row begin delete from comment where comment.u_id=old.id; end|| delimiter ; 有一点很让人郁闷，就是写好的触发器代码，不能修改，你要删除掉重建，郁闷中，对了还有一点就是 phpmyadmin，有的能创建触发器，有的不能，有的能创建，但创建了看不到。在研究一下。 5,测试触发器是否可用 a，测试 updata 触发器
```

update user set name='苍鹰' where id = 1; 更新后去 comment 表里面看看，里面 name 字段里面的段有没有改变 b，测试 delete 触发器

delete from user where id = 1; 更新后去 comment 表里面看看，里面 name 字段里面的段有没有改变 四，触发器的优点 1，触发器的”自动性” 对程序员来说，触发器是看不到的，但是他的确做事情了，如果不用触发器的话，你更新了 user 表的 name 字段时，你还要写代码去更新其他表里面的冗余字段，我举例子，只是一张表，如果是几张表都有冗余字段呢，你的代码是不是要写很多呢，看上去是不是很不爽呢。 2，触发器的数据完整性 触发器有回滚性，举个例子，我发现我很喜欢举子，就是你要更新五张表的数据，不会出现更新了二个张表，而另外三张表没有更新。 但是如果是用 php 代码去写的话，就有可能出现这种情况的，比如你更新了二张表的数据，这个时候，数据库挂掉了。你就郁闷了，有的更新了，有的没更新。

本文由我司收集整编，推荐下载，如有疑问，请与我司联系

这样页面显示不一致了，变有 bug 了。tips:感谢大家的阅读，本文由我司收集整编。仅供参考！